

CLAIMS

What is claimed is:

- 1 1. An apparatus comprising:
2 a key generator to generate an operating system nub key (OSNK) unique to an
3 operating system (OS) nub, the OS nub being part of an operating system running on a
4 secure platform; and
5 a usage protector coupled to the key generator to protect usage of a subset of a
6 software environment using the OSNK.

- 1 2. The apparatus of claim 1 wherein the key generator comprises:
2 a combiner to combine an identification of the OS nub and a master binding key
3 (BK0) of the secure platform, the combined identification and the BK0 corresponding
4 to the OSNK.

- 1 3. The apparatus of claim 2 wherein the identification is a hash value of
2 one of the OS nub and a certificate representing the OS nub.

- 1 4. The apparatus of claim 1 wherein the usage protector comprises:
2 an encryptor to encrypt the subset of the software environment using the OSNK,
3 the encrypted subset being stored in a storage; and
4 a decryptor to decrypt the encrypted subset using the OSNK, the encrypted
5 subset being retrieved from the storage.

- 1 5. The apparatus of claim 1 wherein the usage protector comprises:
2 an encryptor to encrypt a first hash value of the subset of the software
3 environment using the OSNK, the encrypted first hash value being stored in a storage;
4 a decryptor to decrypt the encrypted first hash value using the OSNK, the
5 encrypted first hash value being retrieved from the storage; and
6 a comparator to compare the decrypted first hash value to a second hash value
7 to generate a compared result, the compared result indicating whether the subset of the
8 software environment has been modified.

1 9. The apparatus of claim 1 wherein the secure platform uses an isolated
2 execution mode.

1 10. The apparatus of claim 1 wherein the software environment is one of a
2 Windows operating system, a Windows 95 operating system, a Windows 98 operating
3 system, a Windows NT operating system, and a Windows 2000 operating system.

1 11. The apparatus of claim 1 wherein the subset of the software environment
2 is a registry of an operating system.

1 12. The apparatus of claim 2 wherein the BK0 is generated at random on a
2 first invocation of a processor nub.

1 13. A method comprising:
2 generating an operating system nub key (OSNK) unique to an operating system
3 (OS) nub, the OS nub being part of an operating system running on a secure platform;
4 and
5 protecting usage of a subset of the software environment using the OSNK.

1 14. The method of claim 11 wherein generating the OSNK comprises:
2 combining an identification of the OS nub and a master binding key (BK0) of
3 the secure platform, the combined identification and the BK0 corresponding to the
4 OSNK.

1 15. The method of claim 14 wherein the identification is a hash value of one
2 of the OS nub and a certificate representing the OS nub.

1 16. The method of claim 13 wherein protecting usage comprises:
2 encrypting the subset of the software environment using the OSNK;
3 storing the encrypted subset in a storage; and
4 decrypting the encrypted subset from the storage using the OSNK.

1 17. The method of claim 13 wherein protecting usage comprises:
2 encrypting a first hash value of the subset of the software environment using the
3 OSNK, the encrypted first hash value being stored in a storage;
4 decrypting the encrypted first hash value of the subset of the software
5 environment using the OSNK, the encrypted first hash value being retrieved from the
6 storage; and
7 comparing the decrypted first hash value to a second hash value to generate a
8 compared result, the decrypted first hash value being retrieved from the storage, the
9 compared result indicating whether the subset of the software environment has been
10 modified.

1 18. The method of claim 13 wherein protecting usage comprises:
2 encrypting a first hash value of the subset of the software environment using the
3 OSNK, the encrypted first hash value being stored in a storage;
4 encrypting a second hash value using the OSNK; and
5 comparing the encrypted first hash value to the encrypted second hash value to
6 generate a compared result, the encrypted first hash value being retrieved from the
7 storage, the compared result indicating whether the subset of the software environment
8 has been modified.

1 19. The method of claim 13 wherein protecting usage comprises:
2 decrypting a protected private key to generate a private key using the OSNK;
3 generating a signature of the subset of the software environment using the
4 private key, the signature being stored in a storage; and
5 verifying the signature to generate a modified/not modified flag using a public
6 key, the signature being retrieved from the storage, the modified/not modified flag
7 indicating whether the subset of the software environment has been modified.

1 20. The method of claim 13 wherein detecting comprises:
2 generating a manifest of the subset of the software environment, the manifest
3 describing the subset of the software environment, the manifest being stored in a
4 storage;

5 generating a manifest signature of the manifest using a private key, the private
6 key being decrypted using the OSNK, the manifest signature being stored in the
7 storage;

8 verifying the manifest signature to generate a signature verified flag using a
9 public key, the manifest signature being retrieved from the storage; and

10 verifying the manifest to generate a manifest verified flag, the manifest being
11 retrieved from the storage, the manifest verified flag and the signature verified flag
12 being tested at a test center, the test center generating a pass/fail signal, the pass/fail
13 signal indicating whether the subset of the software environment has been modified.

1 21. The method of claim 13 wherein the secure platform uses an isolated
2 execution mode.

1 22. The method of claim 13 wherein the software environment is one of a
2 Windows operating system, a Windows 95 operating system, a Windows 98 operating
3 system, a Windows NT operating system, and a Windows 2000 operating system.

1 23. The method of claim 13 wherein the subset of the software environment
2 is a registry of the operating system.

1 24. The method of claim 14, wherein the BK0 is generated at random on a
2 first invocation of a processor nub.

1 25. A computer program product comprising:
2 a computer usable medium having computer program code embodied therein,
3 the computer program product having:
4 computer readable program code for generating an operating system nub key
5 (OSNK) unique to an operating system (OS) nub, the OS nub being part of an operating
6 system running on a secure platform; and
7 computer readable program code for protecting usage a subset of the software
8 environment using the OSNK.

1 26. The computer program product of claim 25 wherein the computer
2 readable program code for generating the OSNK comprises:

3 computer readable program code for combining an identification of the OS nub
4 and a master binding key (BK0) of the secure platform, the combined identification and
5 the BK0 corresponding to the OSNK.

1 27. The computer program product of claim 26 wherein the identification is
2 a hash value of one of the OS nub and a certificate representing the OS nub.

1 28. The computer program product of claim 25 wherein the computer
2 readable program code for protecting usage comprises:
3 computer readable program code for encrypting the subset of the software
4 environment using the OSNK;
5 computer readable program code for storing the encrypted subset; and
6 computer readable program code for decrypting the encrypted subset from the
7 storage using the OSNK.

1 29. The computer program product of claim 25 wherein the computer
2 readable program code for protecting usage comprises:
3 computer readable program code for encrypting a first hash value of the subset
4 of the software environment using the OSNK, the encrypted first hash value being
5 stored in a storage;
6 computer readable program code for decrypting the encrypted first hash value
7 of the subset of the software environment using the OSNK, the encrypted first hash
8 value being retrieved from the storage; and
9 computer readable program code for comparing the decrypted first hash value to
10 a second hash value to generate a compared result, the decrypted first hash value being
11 retrieved from the storage, the compared result indicating whether the subset of the
12 software environment has been modified.

1 30. The computer program product of claim 25 wherein the computer
2 readable program code for protecting usage comprises:
3 computer readable program code for encrypting a first hash value of the subset
4 of the software environment using the OSNK, the encrypted first hash value being
5 stored in a storage;

6 computer readable program code for encrypting a second hash value using the
7 OSNK; and
8 computer readable program code for comparing the encrypted first hash value to
9 the encrypted second hash value to generate a compared result, the encrypted first hash
10 value being retrieved from the storage, the compared result indicating whether the
11 subset of the software environment has been modified.

1 31. The computer program product of claim 25 wherein the computer
2 readable program code for protecting usage comprises:
3 computer readable program code for decrypting a protected private key to
4 generate a private key using the OSNK;
5 computer readable program code for generating a signature of the subset of the
6 software environment using the private key, the signature being stored in a storage; and
7 computer readable program code for verifying the signature to generate a
8 modified/not modified flag using a public key, the signature being retrieved from the
9 storage, the modified/not modified flag indicating whether the software environment
10 has been modified.

1 32. The computer program product of claim 25 wherein the computer
2 readable program code for protecting usage comprises:
3 computer readable program code for generating a manifest of the subset of the
4 software environment, the manifest being stored in a storage;
5 computer readable program code for generating a manifest signature of the
6 manifest using a private key, the private key being decrypted using the OSNK, the
7 manifest signature being stored in the storage;
8 computer readable program code for verifying the manifest signature to
9 generate a signature verified flag using a public key, the manifest signature being
10 retrieved from the storage; and
11 computer readable program code for verifying the manifest to generate a
12 manifest verified flag, the manifest being retrieved from the storage, the manifest
13 verified flag and the signature verified flag being tested at a test center, the test center
14 generating a pass/fail signal, the pass/fail signal indicating whether the subset of the
15 software environment has been modified.

1 33. The computer program product of claim 25 wherein the secure platform
2 uses an isolated execution mode.

1 34. The computer program product of claim 25 wherein the software
2 environment is one of a Windows operating system, a Windows 95 operating system, a
3 Windows 98 operating system, a Windows NT operating system, and a Windows 2000
4 operating system.

1 35. The computer program product of claim 25 wherein the subset of the
2 software environment is a registry of an operating system.

1 36. The computer program product of claim 26 wherein the BK0 is
2 generated at random on a first invocation of a processor nub.

1 37. A system comprising:
2 a processor;
3 a storage device coupled to the processor, the storage storing a subset of a
4 software environment; and
5 a usage protector comprising:
6 a key generator to generate an operating system nub key (OSNK) unique to an
7 operating system (OS) nub, the operating system nub being part of a software
8 environment running on a secure platform; and
9 a usage protector coupled to the key generator to protect usage of a subset of the
10 software environment using the OSNK.

1 38. The system of claim 37 wherein the key generator comprises:
2 a combiner to combine an identification of the operating system nub and a
3 master binding key (BK0) of the secure platform, the combined identification and BK0
4 corresponding to the OSNK.

1 39. The system of claim 38 wherein the identification is a hash value of one
2 of the OS nub and a certificate representing the OS nub.

1 40. The system of claim 37 wherein the usage protector comprises:
2 an encryptor to encrypt the subset of the software environment using the OSNK,
3 the encrypted subset being stored in a storage; and
4 a decryptor to decrypt the encrypted subset using the OSNK, the encrypted
5 subset being retrieved from the storage.

1 41. The system of claim 37 wherein the usage protector comprises:
2 an encryptor to encrypt a first hash value of the subset of the software
3 environment using the OSNK, the encrypted first hash value being stored in a storage;
4 a decryptor to decrypt the encrypted first hash value using the OSNK, the
5 encrypted first hash value being retrieved from the storage; and
6 a comparator to compare the decrypted first hash value to a second hash value
7 to generate a compared result, the compared result indicating whether the subset of the
8 software environment has been modified.

1 42. The system of claim 37 wherein the usage protector comprises:
2 a first encryptor to encrypt a first hash value of the subset of the software
3 environment using the OSNK, the encrypted first hash value being stored in a storage;
4 a second encryptor to encrypt a second hash value using the OSNK; and
5 a comparator to compare the encrypted second hash value to the encrypted first
6 hash value to generate a compared result, the encrypted first hash value being retrieved
7 from the storage, the compared result indicating whether the subset of the software
8 environment has been modified.

1 43. The system of claim 37 wherein the usage protector comprises:
2 a decryptor to decrypt a protected private key to generate a private key using the
3 OSNK;
4 a signature generator coupled to the decryptor to generate a signature of the
5 subset of the software environment using the private key, the signature being stored in a
6 storage; and
7 a signature verifier to verify the signature to generate a modified/not modified
8 flag using a public key, the signature being retrieved from the storage, the modified/not

9 modified flag indicating whether the subset of the software environment has been
10 modified.

1 44. The system of claim 37 wherein the usage protector comprises:
2 a manifest generator to generate a manifest of the subset of the software
3 environment, the manifest describing the subset of the software environment, the
4 manifest being stored in a storage;
5 a signature generator coupled to the manifest generator to generate a manifest
6 signature of the manifest using a private key, the private key being decrypted using the
7 OSNK, the manifest signature being stored in the storage;
8 a signature verifier to verify the manifest signature to generate a signature
9 verified flag using a public key, the manifest signature being retrieved from the storage;
10 and
11 a manifest verifier to verify the manifest to generate a manifest verified flag, the
12 manifest being retrieved from the storage, the manifest verified flag and the signature
13 verified flag being tested by a test center, the test center generating a pass/fail signal
14 indicating whether the subset has been modified.

1 45. The system of claim 37 wherein the secure platform uses an isolated
2 execution mode.

1 46. The system of claim 37 wherein the software environment is one of a
2 Windows operating system, a Windows 95 operating system, a Windows 98 operating
3 system, a Windows NT operating system, and a Windows 2000 operating system.

1 47. The system of claim 37 wherein the subset of the software environment
2 is a registry of an operating system.

1 48. The system of claim 38 wherein the BK0 is generated at random on a
2 first invocation of a processor nub.